ELSEVIER

Theoretical Computer Science 137 (1995) 85-108

Theoretical Computer Science

On aggregating teams of learning machines

Sanjay Jain[a,*], Arun Sharma[b]

[a] Department of Information Systems and Computer Science, National University of Singapore, Singapore 0511, Singapore

[b] School of Computer Science and Engineering, The University of New South Wales, Sydney, NSW 2052, Australia

Abstract

A team of learning machines is a multiset of learning machines. A team is said to be successful just in case each member of some nonempty subset of the team is successful. The ratio of the number of machines required to be successful to the size of the team is referred to as the *success ratio* of the team. The present paper investigates for which success ratios can a team be replaced by a single machine without any loss in learning power. The answer depends on the concepts being learned and the criteria of success employed. For a given criterion of success, the minimum cut-off ratio where a team can be replaced by a single machine is referred to as the *aggregation ratio* of the criterion.

The main results in the present paper concern aggregation ratios for vacillatory identification of languages from texts. According to this criterion of success, a learning machine is successful just in case it eventually vacillates between a finite set of grammars instead of converging to a single grammar. For a positive integer n, a machine is said to $\mathbf{TxtFex}_n$-identify a language L just in case the machine converges to up to n grammars for L on any text for L. For such identification criteria, the aggregation ratio is derived for the case $n=2$. It is shown that the collection of languages that can be $\mathbf{TxtFex}_2$-identified by teams with success ratio greater than $\frac{5}{6}$ are the same as those collections of languages that can be $\mathbf{TxtFex}_2$-identified by a single machine. It is also established that $\frac{5}{6}$ is indeed the cut-off point by showing that there are collections of languages that can be $\mathbf{TxtFex}_2$-identified by a team employing six machines, at least five of which are required to be successful, but cannot be $\mathbf{TxtFex}_2$-identified by any single machine. Additionally, aggregation ratios are also derived for finite identification of languages from positive data and for numerous criteria involving language learning from both positive and negative data.

1. Introduction

The present paper investigates the problem of aggregating a team of learning machines into a single learning machine. In other words, we are interested in finding

* Corresponding author. E-mail: sanjay@iscs.nus.sg.

SSDI 0304-3975(94)00162-6

when a team of learning machines can be replaced by a single machine without any loss in learning power.

A team of learning machines is essentially a multiset of learning machines. A team is said to successfully learn a concept just in case each member of some nonempty subset of the team learns the concept. If the size of a team is n and if at least m machines in the team are required to be successful for the team to be successful, then the ratio m/n is referred to as the *success ratio* of the team. The present paper addresses the problem, "For what success ratios can a team be replaced by a single machine without any loss in learning power?" We are especially interested in finding the "minimum cut-off" ratio such that teams with success ratios greater than this cut-off can be simulated by a single machine. Such a cut-off, referred to as *aggregation ratio*, depends on the kind of concepts being learned and the type of success criteria employed. For the problem of learning recursive functions from graphs, the answer is known for the three popularly investigated criteria of success, namely, **Fin** (finite identification), **Ex** (identification in the limit) and **Bc** (behaviorally correct identification). For both **Ex** and **Bc**, Pitt and Smith [23] showed the aggregation ratio to be $\frac{1}{2}$. For finite function identification, **Fin**, it was reported in [15] that the aggregation ratio is $\frac{2}{3}$ (this result can also be argued from a result of Freivalds [12] about probabilistic finite function identification).

The present paper describes aggregation results about language identification from positive data. The main results are in the context of vacillatory identification. To facilitate discussion of these results, we informally present some preliminaries from theory of language learning next.

Languages are sets of sentences and a sentence is a finite object; the set of all possible sentences can be coded into $\mathbb{N}$ – the set of natural numbers. Hence, languages may be construed as subsets of $\mathbb{N}$. A grammar for a language is a set of rules that accepts (or equivalently, generates) the language (see [14]). Essentially, any computer program may be viewed as a grammar. Languages for which a grammar exists are called *recursively enumerable.*

A *text* for a language L is any finite sequence that lists all and only the elements of L; repetitions are permitted. A learning machine is an algorithmic device that outputs grammars on finite initial sequences of texts. Two well studied criteria for a machine to successfully learn a language are *identification in the limit* and *behaviorally correct identification.* We next give an informal definition of these criteria.

A learning machine $\mathbf{M}$ is said to **TxtEx**-identify a language L just in case $\mathbf{M}$, fed any text for L, converges to a correct grammar for L. This is essentially the seminal notion of identification in the limit introduced by Gold [13] (see also [7, 21]).

A learning machine $\mathbf{M}$ is said to **TxtBc**-identify L just in case $\mathbf{M}$, fed any text for L, outputs an infinite sequence of grammars such that after a finite number of incorrect guesses, $\mathbf{M}$ outputs only grammars for L. This criterion was first studied by Case and Lynes [7] and Osherson and Weinstein [21], and is also referred to as "extensional" identification.

Osherson et al. [20] first observed that for **TxtEx**-identification, a team can be aggregated if its success ratio is greater than $\frac{2}{3}$. Hence, in matters of aggregation, identification in the limit of languages from positive data turns out to be similar to finite function identification. On the other hand, a result from [22] can easily be used to show that for **TxtBc**-identification the aggregation ratio is $\frac{1}{2}$. Thus, **TxtEx** and **TxtBc** exhibit different behavior with respect to aggregation.

We now present two more criteria of successful language learning, namely, finite identification and vacillatory identification.

A machine **M** is said to **TxtFin**-identify a language L just in case **M**, fed any text for L, outputs only one grammar and that grammar is for L.[1]

We show that for **TxtFin**-identification, the aggregation ratio is $\frac{2}{3}$. Thus, **TxtFin**-identification shows similar behavior as **TxtEx**-identification and finite function identification so far as aggregation is concerned.

We next consider vacillatory identification of languages from texts in which a machine is required to converge to a finite set of grammars. This notion was studied by Osherson and Weinstein [21] and by Case [5]. It should be noted that in the context of function learning, vacillatory identification turns out to be the same as identification in the limit. This was first shown by Barzdin and Podnieks [2] (see also [8]).

Let n be a positive integer. A learning machine **M** is said to $\mathbf{TxtFex}_n$-identify a language L just in case **M**, fed any text for L, converges in the limit to a finite set, with cardinality $\leqslant n$, of grammars for L. In other words, for any text T for L, there exists a set D of grammars of L, cardinality of $D \leqslant n$, such that **M**, fed T, outputs, after a finite number of incorrect guesses, only grammars from D.

If the upper bound n in $\mathbf{TxtFex}_n$-identification is not specified and the only requirement is that the machine converge to some finite set of grammars for the language, then the criteria is referred to as $\mathbf{TxtFex}_*$-identification.

We show that for $\mathbf{TxtFex}_*$-identification, the aggregation ratio is $\frac{1}{2}$. It is interesting to note that in matters of aggregation $\mathbf{TxtFex}_*$-identification behaves more like **TxtBc**-identification than like **TxtEx**-identification. The problem of aggregation for $\mathbf{TxtFex}_n$, however, turns out to be more difficult. We are able to answer this question for the $n=2$ case, by showing that for $\mathbf{TxtFex}_2$-identification, the aggregation ratio is $\frac{5}{6}$. We establish this by showing that the collections of languages that can be $\mathbf{TxtFex}_2$-identified by teams with success ratios greater than $\frac{5}{6}$ are exactly the same as those collections of languages that can be $\mathbf{TxtFex}_2$-identified by a single machine. Our proof of this result involves a fairly complicated simulation argument. We also establish that $\frac{5}{6}$ is indeed the cut-off point for $\mathbf{TxtFex}_2$ aggregation by employing a diagonalization argument to show that there are

[1] More formally, we allow the machine to output a symbol $\perp$ (denoting "no conjecture yet") on some initial segment of the text and then it will be required to output a correct grammar for the remainder of the text. This is only for technical convenience as it makes the learning machine total and simplifies the proofs.

collections of languages that can be $\mathbf{TxtFex}_2$-identified by a team of six machines, at least five of which are required to be successful, but cannot be $\mathbf{TxtFex}_2$-identified by any single machine.

The problem of aggregation becomes somewhat more manageable if we are prepared to allow the aggregated machine to converge to extra number of grammars. In fact we are able to show that aggregation can be achieved at success ratios just above $\frac{1}{2}$ if the aggregated machine is allowed to converge to extra number of grammars. For example, for any positive integer i, all the collections of languages that can be **TxtEx**-identified by teams of $2i+1$ machines, at least $i+1$ of which are required to be successful, can also be $\mathbf{TxtFex}_{i+1}$-identified by a single machine. More generally, using a fairly straight simulation argument, it can be shown that all the collections of languages that can be $\mathbf{TxtFex}_j$-identified by teams of $2i+1$ machines, at least $i+1$ of which are required to be successful, can also be $\mathbf{TxtFex}_{(i+1)\cdot j}$-identified by a single machine.

In Section 3.7, we show that aggregation ratios for language identification from both positive and negative data follow a pattern similar to function learning.

Before we undertake a formal treatment of issues discussed above, it is useful to motivate the notion of team learning and aggregation. We present a scenario modeled by team identification in the limit of languages from [16].

Consider a situation in which two countries, A and B, are at war with each other. Country B uses a secret language to transmit movement orders to its troops. Country A, with an intention to confuse the troops of country B, wishes to learn a grammar for country B's secret language so that it can transmit conflicting troop movement instructions in that secret language. To accomplish this task, country A employs a team of language learners, each of which perform the following three tasks in a loop:

(a) receive and examine strings of country B's secret language;
(b) guess a grammar for the language whose strings are being received;
(c) transmit conflicting messages based on the grammar guessed in step 2 (so that B's troops think that these messages are from B's Generals).

If one or more of the learners in the team is actually, but possibly unknowingly, successful in correctly learning a grammar for country B's secret language, then country A achieves its purpose of confusing the troops of country B. Of course, the notion of team identification models only part of the above scenario, as issues related to learners transmitting messages back are ignored. However, this scenario illustrates situations in which it is not essential to know which members in the team are successful so far as some are. Answering the question of aggregation ratio in this scenario could tell us under what conditions employing a team and requiring a certain fraction of the team to be successful may not yield any extra learning ability over employing a single machine.

We now proceed formally. Section 2 records the notation and describes preliminary notions and definitions from inductive inference literature. Our results are presented in Section 3.

2. Preliminaries

2.1. Notation

Any unexplained recursion theoretic notation is from [25]. The symbol $\mathbb{N}$ denotes the set of natural numbers $\{0, 1, 2, 3, \ldots\}$. The symbol $\mathbb{N}^+$ denotes the set of positive natural numbers $\{1, 2, 3, \ldots\}$. Unless otherwise specified, $i, j, k, l, m, n, q, r, s, t, x, y$, with or without decorations,[2] range over $\mathbb{N}$. Symbols $\emptyset$, $\subseteq$, $\subset$, $\supseteq$, and $\supset$ denote empty set, subset, proper subset, superset, and proper superset, respectively. Symbols A and S, with or without decorations, range over sets of natural numbers. D, P, Q and X, with or without decorations, range over finite sets. Cardinality of a set S is denoted by $\mathrm{card}(S)$. We say that $\mathrm{card}(A) \leqslant *$ to mean that $\mathrm{card}(A)$ is finite. Intuitively, the symbol, $*$, denotes "finite without any prespecified bound". The letters a and b, with or without decorations, range over $\mathbb{N} \cup \{*\}$. The maximum and minimum of a set are denoted by $\max(\cdot)$, $\min(\cdot)$, respectively, where $\max(\emptyset) = 0$ and $\min(\emptyset)$ is undefined.

Letters f, g, h, and G, with or without decorations, range over *total* functions with arguments and values from $\mathbb{N}$. Symbol $\mathcal{R}$ denotes the set of all total computable functions. $\mathcal{C}$ and $\mathcal{S}$, with or without decorations, range over subsets of $\mathcal{R}$. A pair $\langle i, j \rangle$ stands for an arbitrary, computable, one-to-one encoding of all pairs of natural numbers onto $\mathbb{N}$ (see [25]). Similarly, we can define $\langle \cdot, \ldots, \cdot \rangle$ for encoding multiple tuples of natural numbers onto $\mathbb{N}$. By φ we denote a fixed *acceptable* programming system for the partial computable functions: $\mathbb{N} \to \mathbb{N}$ (see [24, 25, 19]). By φ_i we denote the partial computable function computed by program i in the φ-system. The letter p, in some contexts, with or without decorations, ranges over programs; in other contexts p ranges over total functions with its range being construed as programs. By Φ we denote an arbitrary fixed Blum complexity measure (see [3]) for the φ-system. By W_i we denote $\mathrm{domain}(\varphi_i)$. W_i is, then, the r.e. set/language ($\subseteq N$) accepted (or equivalently, generated) by the φ-program i. Symbol $\mathcal{E}$ will denote the set of all r.e. languages. Symbol L, with or without decorations, ranges over $\mathcal{E}$. Symbol $\mathcal{L}$, with or without decorations, ranges over subsets of $\mathcal{E}$. We denote by $W_{i,s}$ the set $\{x \leqslant s \mid \Phi_i(x) < s\}$. A language L is said to be single valued total iff there exists an f, such that $L = \{\langle x, f(x) \rangle \mid x \in \mathbb{N}\}$. In this case L is also said to be representing (or derived from) f. The quantifiers $\overset{\infty}{\forall}$ and $\overset{\infty}{\exists}$ mean "for all but finitely many" and "there exist infinitely many", respectively.

2.2. Learning machines

We first consider function learning machines.

We assume, without loss of generality, that the graph of a function is fed to a machine in canonical order. For $f \in \mathcal{R}$ and $n \in \mathbb{N}$, we let $f[n]$ denote the finite initial

[2] Decorations are subscripts, superscripts and the like.

segment $\{(x, f(x)) \mid x < n\}$. Clearly, $f[0]$ denotes the empty segment. SEG denotes the set of all finite initial segments, $\{f[n] \mid f \in \mathcal{R} \wedge n \in \mathbb{N}\}$.

Definition 1 (*Gold* [13]). A *function learning machine* is an algorithmic device which computes a mapping from SEG into $\mathbb{N}$.

We now consider language learning machines. A *sequence* σ is a mapping from an initial segment of $\mathbb{N}$ into $(\mathbb{N} \cup \{\#\})$. The *content* of a sequence σ, denoted content (σ), is the set of natural numbers in the range of σ. The *length* of σ, denoted by $|\sigma|$, is the number of elements in σ. For $n \leqslant |\sigma|$, the initial sequence of σ of length n is denoted by $\sigma[n]$. Intuitively, #'s represent pauses in the presentation of data. We let σ, τ and γ, with or without decorations, range over finite sequences. SEQ denotes the set of all finite sequences.

Definition 2. A *language learning machine* is an algorithmic device which computes a mapping from SEQ into $\mathbb{N}$.

The set of all finite initial segments, SEG, can be coded onto $\mathbb{N}$. Also, the set of all finite sequences of natural numbers and #'s, SEQ, can be coded onto $\mathbb{N}$. Thus, in both Definitions 1 and 2, we can view these machines as taking natural numbers as input and emitting natural numbers as output. Henceforth, we will refer to both function learning machines and language learning machines as just learning machines, or simply as machines. We let **M**, with or without decorations, range over learning machines.

It should be noted that for all the identification criteria discussed in this paper, we are assuming, without loss of generality, that the learning machines are total.

2.3. Criteria of learning

2.3.1. Function learning

Finite function identification. For finite function identification only, we assume our learning machines to compute a mapping from SEG into $\mathbb{N} \cup \{\perp\}$. The output of machine **M** on evidential state σ will be denoted by $\mathbf{M}(\sigma)$, where "$\mathbf{M}(\sigma) = \perp$" denotes that **M** does not issue any hypothesis on σ.

Definition 3. **M** **Fin**-*identifies* f (written: $f \in \mathbf{Fin}(\mathbf{M})) \Leftrightarrow (\exists i \mid \varphi_i = f)$ $(\exists n_0)[(\forall n \geqslant n_0)$ $[\mathbf{M}(f[n]) = i] \wedge (\forall n < n_0)[\mathbf{M}(f[n] = \perp]])$. We define the class $\mathbf{Fin} = \{\mathcal{S} \subseteq \mathcal{R} \mid (\exists \mathbf{M})[\mathcal{S} \subseteq \mathbf{Fin}(\mathbf{M})\}$.

Function identification in the limit

Definition 4. (*Gold* [13]). **M** **Ex**-identifies f (written: $f \in \mathbf{Ex}(\mathbf{M})) \Leftrightarrow (\exists i \mid \varphi_i = f)$ $(\overset{\infty}{\forall} n)[\mathbf{M}(f[n]) = i]$. We define the class $\mathbf{Ex} = \{\mathscr{S} \subseteq \mathscr{R} \mid (\exists \mathbf{M})[\mathscr{S} \subseteq \mathbf{Ex}(\mathbf{M})]\}$.

Behaviorally correct function identification.

Definition 5 (*Case and Smith* [8]). **M** **Bc**-identifies f (written: $f \in \mathbf{Bc}(\mathbf{M})) \Leftrightarrow$ $(\overset{\infty}{\forall} n)[\varphi_{\mathbf{M}(f[n])} = f]$. We define the class $\mathbf{Bc} = \{\mathscr{S} \subseteq \mathscr{R} \mid (\exists \mathbf{M})[\mathscr{S} \subseteq \mathbf{Bc}(\mathbf{M})]\}$.

The following proposition summarizes the relationship between the various function learning criteria.

Proposition 1 (*Case and Smith* [8], *Barzdin* [1]). $\mathbf{Fin} \subset \mathbf{Ex} \subset \mathbf{Bc}$.

2.3.2. Language learning

A *text* T for a language L is a mapping from $\mathbb{N}$ into $(\mathbb{N} \cup \{\#\})$ such that L is the set of natural numbers in the range of T. The *content* of a text T, denoted content (T), is the set of natural numbers in the range of T. $T[n]$ denotes the finite initial sequence of T with length n. We say that **M** on T converges (written: $\mathbf{M}(T)\downarrow$) iff $(\exists i)(\overset{\infty}{\forall} n)[\mathbf{M}(T[n]) = i]$. Otherwise **M** is said to diverge on T (written: $\mathbf{M}(T)\uparrow$). If $\mathbf{M}(T)\downarrow$ then we define $\mathbf{M}(T)$ to be the unique i such that $(\overset{\infty}{\forall} n)[\mathbf{M}(T[n]) = i]$.

Finite language identification. Again as in the case of finite function identification, we assume our learning machines to compute a mapping from SEQ into $\mathbb{N} \cup \{\perp\}$. This assumption is for this definition only.

Definition 6. **M** **TxtFin**-*identifies* L (written: $L \in \mathbf{TxtFin}(\mathbf{M})) \Leftrightarrow (\forall$ texts T for $L)$ $(\exists i \mid W_i = L)\,(\exists n_0)[(\forall n \geq n_0)[\mathbf{M}(T[n]) = i] \wedge (\forall n < n_0)[\mathbf{M}(T[n]) = \perp]]$. We define the class $\mathbf{TxtFin} = \{\mathscr{L} \subseteq \mathscr{E} \mid (\exists \mathbf{M})[\mathscr{L} \subseteq \mathbf{TxtFin}(\mathbf{M})]\}$.

Language identification in the limit.

Definition 7 (*Gold* [13]). **M** **TxtEx**-identifies L (written: $L \in \mathbf{TxtEx}(\mathbf{M})) \Leftrightarrow (\forall$ texts T for $L)$ $(\exists i \mid W_i = L)$ $(\overset{\infty}{\forall} n)[\mathbf{M}(T[n]) = i]$. We define the class $\mathbf{TxtEx} = \{\mathscr{L} \subseteq \mathscr{E} \mid (\exists \mathbf{M})[\mathscr{L} \subseteq \mathbf{TxtEx}(\mathbf{M})]\}$.

Behaviorally correct language identification.

Definition 8 (*Osherson and Weinstein* [21], *Case and Lynes* [7]). **M** **TxtBc**-identifies L (written: $L \in \mathbf{TxtBc}(\mathbf{M})) \Leftrightarrow (\forall$ texts T for $L)$ $(\overset{\infty}{\forall} n)[W_{\mathbf{M}(T[n])} = L]$. We define the class $\mathbf{TxtBc} = \{\mathscr{L} \subseteq \mathscr{E} \mid (\exists \mathbf{M})[\mathscr{L} \subseteq \mathbf{TxtBc}(\mathbf{M})]\}$.

Vacillatory language identification. We now introduce the notion of a learning machine finitely converging on a text. Let $\mathbf{M}$ be a learning machine and T be a text. $\mathbf{M}(T)$ *finitely-converges* (written: $\mathbf{M}(T)\Downarrow) \Leftrightarrow \{\mathbf{M}(\sigma) \mid \sigma \subset T\}$ is finite, otherwise we say that $\mathbf{M}(T)$ *finitely-diverges* (written: $\mathbf{M}(T)\Uparrow$). If $\mathbf{M}(T)\Downarrow$, then $\mathbf{M}(T)$ is defined $= \{i \mid (\overset{\infty}{\exists}\sigma \subset T)[\mathbf{M}(\sigma)=i]\}$.

Definition 9 (*Osherson and Weinstein* [21], *Case* [5]). Let $b \in \mathbb{N}^+ \cup \{*\}$. $\mathbf{M}$ $\mathbf{TxtFex}_b$-*identifies* L (written: $L \in \mathbf{TxtFex}_b(\mathbf{M})) \Leftrightarrow (\forall$ texts T for $L)$ $(\exists P \mid \mathrm{card}(P) \leqslant b \wedge (\forall i \in P)[W_i = L])[\mathbf{M}(T)\Downarrow \wedge \mathbf{M}(T) = P]$. We define the class $\mathbf{TxtFex}_b = \{\mathscr{L} \subseteq \mathscr{E} \mid (\exists \mathbf{M})[\mathscr{L} \subseteq \mathbf{TxtFex}_b(\mathbf{M})]\}$.

The following proposition summarizes the relationship between the various language learning criteria.

Proposition 2 (*Osherson and Weinstein* [21], *Case and Lynes* [7], *Case* [5]). $\mathbf{TxtFin} \subset \mathbf{TxtEx} = \mathbf{TxtFex}_1 \subset \mathbf{TxtFex}_2 \subset \cdots \subset \mathbf{TxtFex}_* \subset \mathbf{TxtBc}$.

2.4. Team learning

A team of learning machines is essentially a multiset of learning machines. Definition 10 introduces team learning of functions and Definition 11 introduces team learning of languages.

Definition 10 (*Smith* [26], *Osherson et al.* [20]). Let $\mathbf{I} \in \{\mathbf{Fin}, \mathbf{Ex}, \mathbf{Bc}\}$ and $m, n \in \mathbb{N}^+$.

(a) A team of n machines, $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n$, is said to $\mathbf{Team}_n^m\,\mathbf{I}$-*identify* f (written: $f \in \mathbf{Team}_n^m\mathbf{I}(\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n)$) just in case there exist m **distinct** numbers $i_1, i_2, \ldots, i_m$, $1 \leqslant i_1 < i_2 < \cdots < i_m \leqslant n$, such that each of $\mathbf{M}_{i_1}, \mathbf{M}_{i_2}, \ldots, \mathbf{M}_{i_m}$ $\mathbf{I}$-identifies f.

(b) $\mathbf{Team}_n^m\mathbf{I} = \{\mathscr{S} \subseteq \mathscr{R} \mid (\exists \mathbf{M}_1, \exists \mathbf{M}_2, \ldots, \exists \mathbf{M}_n)[\mathscr{S} \subseteq \mathbf{Team}_n^m\mathbf{I}(\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n)]\}$.

Definition 11. Let $b \in \mathbb{N}^+ \cup \{*\}$. Let $\mathbf{I} \in \{\mathbf{TxtFin}, \mathbf{TxtEx}, \mathbf{TxtFex}_b, \mathbf{TxtBc}\}$. Let $m, n \in \mathbb{N}^+$.

(a) A team of n machines $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n$ is said to $\mathbf{Team}_n^m\mathbf{I}$-*identify* L (written: $L \in \mathbf{Team}_n^m\mathbf{I}(\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n)$) just in case there exist m distinct numbers $i_1, i_2, \ldots, i_m$, $1 \leqslant i_1 < i_2 < \cdots < i_m \leqslant n$, such that each of $\mathbf{M}_{i_1}, \mathbf{M}_{i_2}, \ldots, \mathbf{M}_{i_m}$ $\mathbf{I}$-identifies L.

(b) $\mathbf{Team}_n^m\mathbf{I} = \{\mathscr{L} \subseteq \mathscr{E} \mid (\exists \mathbf{M}_1, \exists \mathbf{M}_2, \ldots, \exists \mathbf{M}_n)\ [\mathscr{L} \subseteq \mathbf{Teasm}_n^m\mathbf{I}(\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n)]\}$.

For $\mathbf{Team}_n^m\mathbf{I}$-identification criteria, we refer to the fraction m/n as the *success ratio* of the criteria.

Definition 12. A reduced fraction m/n is referred to as the *aggregation ratio* for the success criteria $\mathbf{I}$-identification just in case

(a) $(\forall i, j \in \mathbb{N}^+ \mid i/j > m/n)[\mathbf{Team}_j^i\mathbf{I} = \mathbf{I}]$,

(b) $\mathbf{I} \subset \mathbf{Team}_n^m\mathbf{I}$.

If the aggregation ratio for **I**-identification is m/n, then aggregation for **I**-identification takes place at success ratios greater than m/n. Additionally, m/n is indeed the cut-off point of aggregation for **I**-identification.

In the following, for $i>j$, we take $\mathbf{Team}^i_j\mathbf{I}=\{\emptyset\}$.

3. Results

3.1. Previously known results

Aggregation results are known for all the function learning criteria defined in the previous section. For finite function identification, the aggregation ratio is $\frac{2}{3}$. This is implied by the following theorem, part (a) of which appeared in [15] and can also easily be argued from a related result of Freivalds [12] about probabilistic finite identification. Theorem 1(b) can be established via a diagonalization argument employing the operator recursion theorem [4].

Theorem 1 (Velauthapillai [27], Jain and Sharma [15]). (a) $(\forall m,n\in\mathbb{N}^+ \mid m/n>\frac{2}{3})$ $[\mathbf{Team}^m_n\mathbf{Fin}=\mathbf{Fin}]$.

(b) $\mathbf{Fin}\subset\mathbf{Team}^2_3\mathbf{Fin}$.

Pitt and Smith [23] settled the question for function identification in the limit and behaviorally correct function identification by showing the following theorem which implies that for both these criteria the aggregation ratio is $\frac{1}{2}$.

Theorem 2 (Pitt and Smith [23], Smith [26]). *Let* $\mathbf{I}\in\{\mathbf{Ex},\mathbf{Bc}\}$. (a) $(\forall m,n\in\mathbb{N}^+ \mid m/n>\frac{1}{2})$ $[\mathbf{Team}^m_n\mathbf{I}=\mathbf{I}]$.

(b) $\mathbf{I}\subset\mathbf{Team}^1_2\mathbf{I}$.

For language learning, the result is known for **TxtEx**-identification and **TxtBc**-identification. It was shown by Osherson et al. [20] that aggregation ratio for **TxtEx**-identification is $\frac{2}{3}$ (see also [16–18] for extension of this result to anomalies in the final grammar).

Theorem 3 (Osherson et al. [20]). (a) $(\forall m,n\in\mathbb{N}^+ \mid m/n>\frac{2}{3})[\mathbf{Team}^m_n\mathbf{TxtEx}=\mathbf{TxtEx}]$.

(b) $\mathbf{TxtEx}\subset\mathbf{Team}^2_3\mathbf{TxtEx}$.

The next theorem implies that the aggregation ratio for **TxtBc** is $\frac{1}{2}$. Theorem 4(a) follows from a result of Pitt [22], and part (b) of Theorem 4 can be proved by considering a collection of single valued total languages derived from the corresponding function learning result of Smith (Theorem 2(b)).

Theorem 4. (a) $(\forall m, n \in \mathbb{N}^+ \mid m/n > \frac{1}{2})[\mathbf{Team}_n^m\mathbf{TxtBc} = \mathbf{TxtBc}]$.
(b) $\mathbf{TxtBc} \subset \mathbf{Team}_2^1\mathbf{TxtBc}$.

We now consider aggregation for **TxtFin**-identification and $\mathbf{TxtFex}_b$-identification, $b \in \mathbb{N}^+ \cup \{*\}$.

3.2. Aggregation for finite identification of languages

It turns out that aggregation for finite identification of languages is no different from aggregation for limit identification of languages. Theorem 5 shows that the aggregation ratio for **TxtFin**-identification is $\frac{2}{3}$. A proof of part (a) can be obtained on the lines of the proof of Theorem 1(a). A proof of part (b) can be worked out by considering the collection of single valued total languages derived from the class of functions considered in the proof of Theorem 1(b).

Theorem 5. (a) $(\forall m, n \mid m/n > \frac{2}{3})[\mathbf{Team}_n^m\mathbf{TxtFin} = \mathbf{TxtFin}]$.
(b) $\mathbf{TxtFin} \subset \mathbf{Team}_3^2\mathbf{TxtFin}$.

3.3. Aggregation for vacillatory identification of languages

In the present section, we consider the problem of aggregation for vacillatory identification of languages. We first introduce some technical machinery that simplifies the description of our proofs.

Definition 13. Let $k \in \mathbb{N}$ and T be a text.
(a) Let $n \in \mathbb{N}$. $\mathrm{Match}(k, T[n]) = \max(\{m \leqslant n \mid \mathrm{content}(T[m]) \subseteq W_{k,n} \wedge W_{k,m} \subseteq \mathrm{content}(T[n])\})$.
(b) $\mathrm{Match}(k, T) = \lim_{n\to\infty} \mathrm{Match}(k, T[n])$ if the limit exists; $\mathrm{Match}(k, T) = \infty$ otherwise.

Intuitively, $\mathrm{Match}(k, T[n])$, measures how much W_k and $T[n]$ agree with each other. Match is employed in the process of determining if a given grammar k is for the language $\mathrm{content}(T)$. The following simple lemma summarizes the properties of Match; its proof is straightforward.

Lemma 1. *Let $k \in \mathbb{N}$ and T be a text.*
(a) *If $W_k = \mathrm{content}(T)$, then $Match(k, T) = \infty$.*
(b) *If $W_k \neq \mathrm{content}(T)$, then $Match(k, T) < \infty$.*

The next definition introduces a function that keeps track of some finite number of grammars output by a machine on the initial segment of a text.

Definition 14. Let $b \in \mathbb{N}^+ \cup \{*\}$. Let $\mathbf{M}$ be a machine and T be a text.
(a) Let $n \in \mathbb{N}$. $\text{LastGram}_b(\mathbf{M}(T[n]) = \{\mathbf{M}(T[m]) \mid \text{card}(\mathbf{M}(T[m']) \mid m \leqslant m' \leqslant n) \leqslant b\}$.
(b) $\text{LastGram}_b(\mathbf{M}, T) = \lim_{n \to \infty} \text{LastGram}_b(\mathbf{M}(T[n])$ ($\text{LastGram}_b(\mathbf{M}, T)$ is undefined if the limit does not exist).

Intuitively, for $b \in \mathbb{N}$, $\text{LastGram}_b(\mathbf{M}, T[n])$ is the set of last b distinct grammars output by $\mathbf{M}$ on initial segments of $T[n]$. Last $\text{Gram}_*(\mathbf{M}(T[n]))$ is the set of all distinct grammars output by $\mathbf{M}$ on initial segments of $T[n]$.

The next definition introduces a function that keeps track of the point in the initial segments of text where a machine undergoes a mind change with respect to $\mathbf{TxtFex}_b$-identification.

Definition 15. Let $b \in \mathbb{N}^+ \cup \{*\}$, $\mathbf{M}$ be a machine and T be a text.
(a) Let $n \in \mathbb{N}$. $\text{LastMindChange}_b(\mathbf{M}, T[n]) = \max(\{m < n \mid \text{LastGram}_b(\mathbf{M}, T[m]) \neq \text{LastGram}_b(\mathbf{M}, T[m+1])\})$.
(b) $\text{LastMindChange}_b(\mathbf{M}, T) = \lim_{n \to \infty} \text{LastMindChange}_b(\mathbf{M}, T[n])$ if the limit exists; $\text{LastMindChange}_b(\mathbf{M}, T) = \infty$ otherwise.

So, $\text{LastMindChange}_b(\mathbf{M}, T)$ computes the last point in the text T where machine $\mathbf{M}$ undergoes a mind change with respect to $\mathbf{TxtFex}_b$-identification.

Finally we define the following definition.

Definition 16. Let S be a nonempty finite subset of $\mathbb{N}$ and T a text. Let $n \in \mathbb{N}$. $\text{BestGram}(S, T[n]) =$ least $i \in S$ such that $\text{Match}(i, T[n])$ is maximized.

So, $\text{BestGram}(S, T[n])$ finds the best candidate grammar for $\text{content}(T)$ from the set of grammars S based on the data available in $T[n]$. The following lemma, whose proof is straightforward, is a useful observation about the function BestGram.

Lemma 2. *Let S be a nonempty finite subset of $\mathbb{N}$ and T a text. If there exists an $i \in S$ such that $W_i = \text{content}(T)$, then for all but finitely many n, $\text{BestGram}(S, T[n])$ is a grammar for* $\text{content}(T)$.

We now present our results.

3.4. *Aggregation for* $\mathbf{TxtFex}_*$

Our first result for team aggregation in the context of vacillatory identification is for $\mathbf{TxtFex}_*$-identification. Theorem 6 says that the aggregation ratio for $\mathbf{TxtFex}_*$-identification is $\frac{1}{2}$. It is interesting to observe that in matters of aggregation, $\mathbf{TxtFex}_*$-identification behaves more like $\mathbf{TxtBc}$-identification than like $\mathbf{TxtEx}$-identification.

Theorem 6. (a) $(\forall i, j \in \mathbb{N}^+ \mid i/j > \frac{1}{2})[\mathbf{Team}_j^i\mathbf{TxtFex}_* = \mathbf{TxtFex}_*]$.
(b) $\mathbf{TxtFex}_* \subset \mathbf{Team}_2^1\mathbf{TxtFex}_*$.

Proof. (a) Let i, j be as given in the hypothesis of the theorem. Suppose a team of j machines, $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_j$, is given. We describe a machine $\mathbf{M}$ such that $\mathbf{Team}_j^i\mathbf{TxtFex}_*(\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_j) \subseteq \mathbf{TxtFex}_*(\mathbf{M})$.

Let S_n be the lexicographically least subset of $\{1, 2, \ldots, j\}$of cardinality i such that $\max(\{\text{LastMindChange}_*(\mathbf{M}_k, T[n]) \mid k \in S_n\})$ is minimized.

$\mathbf{M}(T[n])$ is defined as follows.

$$\mathbf{M}(T[n]) = \text{BestGram}\left(\bigcup_{j \in S_n} \text{LastGram}_*(\mathbf{M}_j, T[n]), T[n]\right).$$

We claim that if $L \in \mathbf{Team}_j^i\mathbf{TxtFex}_*(\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_j)$, then $L \in \mathbf{TxtFex}_*(\mathbf{M})$. To see this suppose T is a text for L. Suppose S is the lexicographically least subset of $\{1, 2, \ldots, j\}$ of cardinality i such that $\max(\{\text{LastMindChange}_*(\mathbf{M}_k, T) \mid k \in S\})$ is minimized. Note that if $k \in S$, then $\mathbf{M}_k$ finitely converges on T. Clearly, $\lim_{n \to \infty} S_n = S$. Also, since $i > \frac{1}{2}j$, there exists $k \in S$, such that $\text{LastGram}_*(\mathbf{M}_k, T)$ contains a grammar for L.

Thus, $\mathbf{M}(T)$ finitely converges and, for large enough n, $\mathbf{M}(T[n])$ is a grammar for L.

(b) For team function learning, we know that $\mathbf{Team}_2^1\mathbf{Ex} - \mathbf{Ex} \neq \emptyset$ [26]. Also, since $\mathbf{Fex} = \mathbf{Ex}$ [2, 8], we have $\mathbf{Team}_2^1\mathbf{Fex} - \mathbf{Fex} \neq \emptyset$. Let $\mathcal{S} \in (\mathbf{Team}_2^1\mathbf{Fex} - \mathbf{Fex})$. Now, it is easy to verify that the collection of single valued total language, representing functions in $\mathcal{S}$, witnesses $\mathbf{Team}_2^1\mathbf{TxtFex}_* - \mathbf{TxtFex}_* \neq \emptyset$. We omit the details. □

3.5. Pseudo-aggregation results

The problem of finding aggregation ratios for $\mathbf{TxtFex}_b$-identification when $b \neq *$ turns out to be far more difficult. The difficulty arises in requiring the aggregated machine to also converge to up to b grammars. In the light of these difficulties, it is worth considering cases where the bound on the number of converged grammars for the aggregated machine is more than the bound allowed for the team. Such a relaxation on aggregation is referred to as "pseudo-aggregation", and representative results are presented next.

It can be shown that $\mathbf{Team}_5^3\mathbf{TxtEx} - \mathbf{TxtFex}_2 \neq \emptyset$, but $\mathbf{Team}_5^3\mathbf{TxtEx} \subseteq \mathbf{TxtFex}_3$. Hence, allowing more grammars in the limit can sometimes help achieve pseudo-aggregation. This result can be generalized to show the following theorem.

Theorem 7. *Let* $i \in \mathbb{N}^+$.
(a) $\mathbf{Team}_{2i+1}^{i+1}\mathbf{TxtEx} - \mathbf{TxtFex}_i \neq \emptyset$.
(b) $\mathbf{Team}_{2i+1}^{i+1}\mathbf{TxtEx} \subseteq \mathbf{TxtFex}_{i+1}$.

The next result generalizes Theorem 7.

Theorem 8. *Let* $i, j \in \mathbb{N}^+$.
(a) $\mathbf{Team}_{2i+1}^{i+1}\mathbf{TxtFex}_j - \mathbf{TxtFex}_{(i+1)\cdot j-1} \neq \emptyset$.
(b) $\mathbf{Team}_{2i+1}^{i+1}\mathbf{TxtFex}_j \subseteq \mathbf{TxtFex}_{(i+1)\cdot j}$.

Proof. A proof similar to the one used to prove Theorem 6(a) can be employed to establish part (b). We give a proof of part (a). Consider the following collection of languages:

$$\mathscr{L} = \{L \in \mathscr{E} \mid \mathrm{card}(\{x \mid \langle 0, x\rangle \in L\}) = (i+1)*j \text{ and}$$
$$\mathrm{card}(\{x \mid \langle 0, x\rangle \in L \wedge W_x = L\}) \in \{1, (i+1)*j\}.$$

We first show that $\mathscr{L} \in \mathbf{Team}_{2i+1}^{i+1}\mathbf{TxtFex}_j$. We describe machines, $\mathbf{M}_1, \ldots, \mathbf{M}_{2i+1}$ which $\mathbf{Team}_{2i+1}^{i+1}\mathbf{TxtFex}_j$-identify $\mathscr{L}$. Suppose T is a text for $L \in \mathscr{L}$. Let $S_n = \{x \mid \langle 0, x\rangle \in \mathrm{content}(T[n])\}$. Let

$$w_n^k = \begin{cases} x & \text{if } \mathrm{card}(S_n) \geqslant k, \text{ and } x \in S_n \text{ and } \mathrm{card}(\{y \leqslant x \mid y \in S_n\}) = k, \\ 0 & \text{otherwise.} \end{cases}$$

So, w_n^k is the kth element in S_n, if any.

For $1 \leqslant k \leqslant i+1$, let $\mathbf{M}_k(T[n]) = \mathrm{BestGram}(\{w_n^{k'} \mid (k-1)*j < k' \leqslant k*j\}, T[n])$. For $i+1 < k \leqslant 2i+1$, let $\mathbf{M}_k(T[n]) = \mathrm{BestGram}(\{w_n^{k'} \mid 0 < k' \leqslant (i+1)*j\}, T[n])$. It is easy to see that, if $\mathrm{card}(\{x \mid \langle 0, x\rangle \in L\} \wedge W_x = L) = (i+1)*j$, then each of $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{i+1}$ $\mathbf{TxtFex}_j$-identify L. On the other hand, if $\mathrm{card}(\{x \mid \langle 0, x\rangle \in L \wedge W_x = L\}) = 1$, then at least one of $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{i+1}$ and each of $\mathbf{M}_{i+2}, \ldots, \mathbf{M}_{2i+1}$ $\mathbf{TxtEx}$-identify L. Thus, $\mathscr{L} \in \mathbf{Team}_{2i+1}^{i+1}\mathbf{TxtFex}_j$.

We now show that $\mathscr{L} \notin \mathbf{TxtFex}_{(i+1)*j-1}$. Suppose by way of contradiction that machine $\mathbf{M}$ $\mathbf{TxtFex}_{(i+1)*j-1}$-identifies $\mathscr{L}$. We then show that there exists a language in $\mathscr{L}$ that $\mathbf{M}$ fails to $\mathbf{TxtFex}_{(i+1)*j-1}$-identify. The description of this witness proceeds in stages and uses the multiple recursion theorem. We first give an informal idea of the construction.

We describe languages accepted by $(i+1)*j$ grammars, $k_1, k_2, \ldots, k_{(i+1)*j}$. At each stage s, the construction makes use of initial sequences σ_s. By the use of $(i+1)*j$-ary recursion theorem, we initialize σ_0 to contain elements $\langle 0, k_1\rangle$. $\langle 0, k_2\rangle, \ldots, \langle 0, k_{(i+1)*j}\rangle$. This step ensures that the languages accepted by these grammars will be members of $\mathscr{L}$. We then proceed in stages. At each stage s, an attempt is made to find a sequence τ extending σ_s such that $\mathbf{M}$ undergoes a mind change on τ with respect to $\mathbf{TxtFex}_{(i+1)*j-1}$-identification. If such an attempt is successful at every stage then each of the grammars $k_1, k_2, \ldots, k_{(i+1)*j}$ will be for the same language and this language will be a member of $\mathscr{L}$. But, $\mathbf{M}$ will fail to converge to a set of up to $(i+1)*j-1$ grammars on a text for this language and hence $\mathbf{M}$ will not $\mathbf{TxtFex}_{(i+1)*j-1}$-identify this language. If on the other hand, an attempt to find a mind change is unsuccessful at some stage then the construction makes sure that each of the grammars $k_1, k_2, \ldots, k_{(i+1)*j}$ are for pairwise distinct languages in $\mathscr{L}$. Not only are these languages pairwise distinct but they are also infinitely different from each other. Now, since the machine **M** gets locked to a set of no more than $(i+1)*j-1$ grammars on some text for each of the $(i+1)*j$ languages, the machine **M** will fail to $\mathbf{TxtFex}_{(i+1)*j-1}$-identify at least one of these languages. We now proceed formally.

By the $(i+1)*j$-ary recursion theorem (see [4]) there exist grammars $k_1, k_2, \ldots, k_{(i+1)*j}$ such that the languages W_{k_s} may be described as follows.

Let σ_0 be a sequence such that $\mathrm{content}(\sigma_0)=\{\langle 0, k_l\rangle \mid 1 \leqslant l \leqslant (i+1)*j\}$. Go to stage 0.

Begin $\{$stage $s\}$

Enumerate content(σ_s) in W_{k_l}, $1 \leqslant l \leqslant (i+1)*j$.

Dovetail steps 1 and 2 below until step 1 succeeds. If and when step 1 succeeds, go to step 3.

1. Search for a $\tau \supseteq \sigma_s$ such that $\mathrm{content}(\tau)-\mathrm{content}(\sigma_s) \subseteq \{\langle x, y\rangle \mid 1 \leqslant x\}$ and $\mathrm{LastGram}_{(i+1)*j-1}(\mathbf{M}, \tau) \neq \mathrm{LastGram}_{(i+1)*j-1}(\mathbf{M}, \sigma_s)$.
2. Let $y=0$.
   Go to substage 0.
   Begin $\{$substage $s'\}$
   Enumerate $\langle l, y\rangle$ in W_{k_l}, for $1 \leqslant l \leqslant (i+1)*j$.
   Let $y=y+1$.
   Go to substage $s'+1$.
   End $\{$substage $s'\}$
3. Let $\sigma_{s+1} \supseteq \tau$ be such that $\mathrm{content}(\sigma_{s+1})=\mathrm{content}(\tau) \cup \bigcup_{1 \leqslant l \leqslant (i+1)*j}[W_{k_l}$ enumerated till now]. Go to stage $s+1$.

End $\{$stage $s\}$.

We now consider the following cases.

Case 1: All stages halt. In this case, let $L=W_{k_1}=W_{k_2}=\cdots=W_{k_{(i+1)*j}} \in \mathscr{L}$. Clearly, $T=\bigcup_s \sigma_s$ is a text for L. However, **M** on T does not finitely converge to a set of $(i+1)*j-1$ grammars.

Case 2: Some stage s starts but does not finish. In this case, let $L_l=W_{k_l}$, for $1 \leqslant l \leqslant (i+1)*j$. Now, clearly $L_l \neq L_{l'}$ for $l \neq l'$, $1 \leqslant l$, $l' \leqslant (i+1)*j$. But on all texts, T, extending σ_s for each L_l, $\mathrm{LastGram}_{(i+1)*j-1}(\mathbf{M}, T)=\mathrm{LastGram}_{(i+1)*j-1}(\mathbf{M}, \sigma_s)$. Since, $\mathrm{LastGram}_{(i+1)*j-1}(\mathbf{M}, \sigma_s)$ has at most $(i+1)*j-1$ grammars, there exists a language in $\{L_l \mid 1 \leqslant l \leqslant (i+1)*j\}$, which **M** does not $\mathbf{TxtFex}_{(i+1)*j-1}$-identify. □

3.6. *Aggregation for* $\mathbf{TxtFex}_2$

The results in the previous section do not say anything about aggregation in the context of $\mathbf{TxtFex}_b$-identification, when $b \neq *$. The following result shows that aggregation ratio for $\mathbf{TxtFex}_2$-identification is $\geqslant \frac{2}{3}$ and aggregation ratio for $\mathbf{TxtFex}_3$-identification is $\geqslant \frac{3}{4}$.

Theorem 9. *Let* $i \in \mathbb{N}^+$. $\mathbf{Team}^i_{i+1}\mathbf{TxtFex}_i - \mathbf{TxtFex}_i \neq \emptyset$.

Proof. We prove this result as a direct consequence of the following lemma.

Lemma 3. $\mathbf{TxtFex}_{i+1} \subseteq \mathbf{Team}^i_{i+1}\,\mathbf{TxtFex}_i$.

Before we give a proof of the lemma, we show how the lemma implies the theorem. Suppose by way of contradiction the theorem is not true. Hence,we have $\mathbf{Team}^i_{i+1}\,\mathbf{TxtFex}_i \subseteq \mathbf{TxtFex}_i$. This, together with the lemma, implies that $\mathbf{TxtFex}_{i+1} \subseteq \mathbf{Team}^i_{i+1}\,\mathbf{TxtFex}_i \subseteq \mathbf{TxtFex}_i$. But, this yields $\mathbf{TxtFex}_i = \mathbf{TxtFex}_{i+1}$ – a contradiction.

We now give a proof of the lemma. Suppose $\mathbf{M}$ is given. We describe $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{i+1}$ such that $\mathbf{TxtFex}_{i+1}(\mathbf{M}) \subseteq \mathbf{Team}^i_{i+1}\,\mathbf{TxtFex}_i(\mathbf{M}_1, \ldots, \mathbf{M}_{i+1})$.

Suppose T is a text for $L \in \mathbf{TxtFex}_{i+1}(\mathbf{M})$. Let $S_n = \mathrm{LastGram}_{i+1}(\mathbf{M}, T[n])$. Let the elements of S_n be $w_n^1 < w_n^2 < \cdots < w_n^{\mathrm{card}(S_n)}$. For $\mathrm{card}(S_n) < l \leqslant i+1$, let $w_n^l = l + \max(S_n)$. For $1 \leqslant k \leqslant i+1$, let $\mathbf{M}_k(T[n]) = \mathrm{BestGram}(S_n - \{w_n^k\}, T[n])$.

Now since $\mathbf{M}$ on T converges to a set of at most $i+1$ grammars, $\lim_{n\to\infty} S_n$ converges to $\mathrm{LastGram}_{i+1}(\mathbf{M}, T)$, and thus for each k, $1 \leqslant k \leqslant i+1$, $\lim_{n\to\infty} w_n^k$ converges to say w^k.

Since $\mathrm{LastGram}_{i+1}(\mathbf{M}, T)$ contains a grammar for L, and since each w^k are distinct, we have

(a) $\mathrm{LastGram}_{i+1}(\mathbf{M}, T) \subseteq \{w^k \mid 1 \leqslant k \leqslant i+1\}$,

(b) For each k, $1 \leqslant k \leqslant i+1$, $\mathrm{card}(\mathrm{LastGram}_{i+1}(\mathbf{M}, T) - \{w^k\}) \leqslant i$, and

(c) for at least i of k's in $\{1, 2, \ldots, i+1\}$, $(\mathrm{LastGram}_{i+1}(\mathbf{M}, T) - \{w^k\})$ contains a grammar for L.

It follows that at least i of $\mathbf{M}_1, \ldots, \mathbf{M}_{i+1}$ $\mathbf{TxtFex}_i$-identify L. This proves the lemma and the theorem. □

Theorem 9 is not optimal. We consider the special case of $i = 2$. We are able to show that $\mathbf{TxtFex}_2$ aggregation takes place for success ratios greater than $\frac{5}{6}$ as implied by Theorems 10 and 11. The proof of Theorem 10 requires the following crucial technical lemma.

Lemma 4. *Suppose $r, w \in \mathbb{N}$ are given such that $r \geqslant w > \frac{2}{5}r$. There exist recursive functions G_1 and G_2 such that, $(\forall p_1, p_2, \ldots, p_r)(\forall L)[\mathrm{card}(\{i \mid 1 \leqslant i \leqslant r \wedge W_{p_i} = L\}) \geqslant w \Rightarrow W_{G_1(p_1, \ldots, p_r)} = L \vee W_{G_2(p_1, \ldots, p_r)} = L]$.*

Proof. We assume without loss of generality that $w \leqslant \frac{1}{2}r$ (otherwise the lemma can be easily proved by considering the grammar which enumerates elements enumerated by majority of $p_1, \ldots, p_r$).

Suppose $p_1, \ldots, p_r$ are given (we assume, without loss of generality, that they are pairwise distinct). Below, we give a procedure to enumerate two languages L_1 and L_2 (the procedure depends on $p_1, \ldots, p_r$). We will then argue that

$$(\forall L)[\mathrm{card}(\{i \mid 1 \leqslant i \leqslant r \wedge W_{p_i} = L\}) \geqslant w \Rightarrow L = L_1 \vee L = L_2].$$

It will be easy to see that grammars for L_1 and L_2 can be obtained effectively from $p_1, \ldots, p_r$. This will prove the lemma.

The idea of the proof is that, in successive stages, we try to construct two disjoint groups of grammars (from $p_1, \ldots, p_r$) of size w each. These groupings are done with a view to group "similar" grammars together (i.e., grammars that seem to be for the same language). The groupings eventually become correct. Some care is needed in the construction to guard against initial misgrouping of the grammars. We guarantee this with the help of a number of invariants that are satisfied by the construction at the end of each stage. We now introduce a function that, in some sense, measures the similarity between two grammars.

Definition 17. Let $i,j \in \mathbb{N}$. Let $n \in \mathbb{N}$. $\mathrm{Similar}(i,j,n) = \max(\{n_1 \leqslant n \mid W_{i,n_1} \subseteq W_{j,n} \wedge W_{j,n_1} \subseteq W_{i,n}\})$.

So, $\mathrm{Similar}(i,j,n)$ denotes the point where it appears that the languages accepted by the two grammars differ. Following properties of Similar can easily be verified.

(a) $W_i = W_j \Rightarrow \lim_{n \to \infty} \mathrm{Similar}(i,j,n) = \infty$.

(b) $W_i \neq W_j \Rightarrow \lim_{n \to \infty} \mathrm{Similar}(i,j,n) < \infty$.

(c) Let P be a finite subset of $\mathbb{N}$. Let $n \in \mathbb{N}$. If $m = \min(\{\mathrm{Similar}(i,j,n) \mid i,j \in P\})$ then $\bigcup_{k \in P}[W_{k,m}] \subseteq \bigcap_{i \in P}[W_{i,n}]$.

We now describe the data structure employed by the construction. The languages L_1 and L_2 are enumerated in stages. We let L_1^s and L_2^s denote L_1 and L_2 enumerated before stage s, respectively. Also, $e1_s, e2_s$ will be a permutation of 1, 2 (this is used to make a correct correspondence between the two groups of grammars and the two languages). The two groups of grammars just before the execution of stage s are denoted by $P1_s$ and$P2_s$. $P1_s$ and $P2_s$ will be disjoint subsets of $\{1, \ldots, r\}$ of size w each.

The variables used in the construction are initialized as follows. Let $n_0 = 0$, $m1_0 = m2_0 = 0$. Let $e1_0 = 1$ and $e2_0 = 2$. Let $P1_0 = \{1, \ldots, w\}$ and $P2_0 = \{w+1, \ldots, 2w\}$.

The following invariants are maintained by the construction.

Invariants (assuming that stage s is executed)

(H1) $L_{e1_s}^s = \bigcup_{i \in P1_s}[W_{p_i, m1_s}] \subseteq \bigcap_{i \in P1_s}[W_{p_i, n_s}]$.

(H2) $L_{e2_s}^s \supseteq \bigcup_{i \in P2_s}[W_{p_i, m2_s}]$.

(H3) $\bigcup_{i \in P2_s}[W_{p_i, m2_s}] \subseteq \bigcap_{i \in P2_s}[W_{p_i, n_s}]$.

(H4) $L_{e2_s}^s - \bigcup_{i \in P2_s}[W_{p_i, m2_s}] \subseteq L_{e1_s}^s$.

(H5) $(\forall x \in L_{e2_s}^s)[\mathrm{card}(\{j \in \{1,2,\ldots,r\} - P1_s \mid x \in W_{p_j, n_s}\}) \geqslant \frac{1}{2}w]$.

(H6) $m1_{s+1} > n_s \geqslant m1_s \geqslant m2_s$.

Begin {stage s}

1. Search for $n > n_s$ such that there exist a set $P \subseteq \{1, \ldots, r\}$ of cardinality w such that, for all $i,j \in P$, $\mathrm{Similar}(p_i, p_j, n) > n_s$.

2. If such an n is found, let $n_{s+1}=n$.
3. Let $P1_{s+1}\subseteq\{1,\ldots,r\}$ be of cardinality w such that $m1_{s+1}=\min(\{\text{Similar}(p_i,p_j,n_{s+1})\mid i,j\in P1_{s+1}\})$ is maximized.
4. **if** card$(P1_{s+1}\cap P1_s)>$card$(P1_{s+1}\cap P2_s)$, **then** let $e1_{s+1}=e1_s$ and $e2_{s+1}=e2_s$,
   else let $e1_{s+1}=e2_s$ and $e2_{s+1}=e1_s$.
   endif
5. Let $P2'_{s+1}\subseteq\{1,\ldots,r\}-P1_{s+1}$ be of cardinality w such that $m2'_{s+1}=\min(\{\text{Similar}(p_i,p_j,n_{s+1})\mid i,j\in P2'_{s+1}\})$ is maximized.
6. **if** $[P1_{s+1}\cap P1_s\neq\emptyset\wedge P1_{s+1}\cap P2_s\neq\emptyset]\vee[L^s_{e2_{s+1}}\subseteq\bigcup_{i\in P2'_{s+1}}[W_{p_i,m2'_{s+1}}]]$ **then**
   let $P2_{s+1}=P2'_{s+1}$ and $m2_{s+1}=m2'_{s+1}$.
   elseif $e2_{s+1}=e2_s$ **then** let $P2_{s+1}=P2_s$, $m2_{s+1}=m2_s$.
   else let $P2_{s+1}=P1_s$, $m2_{s+1}=m1_s$.
   endif
7. Enumerate $\bigcup_{i\in P1_{s+1}}[W_{p_i,m1_{s+1}}]$in $L_{e1_{s+1}}$.
   Enumerate $\bigcup_{i\in P2_{s+1}}[W_{p_i,m2_{s+1}}]$ in $L_{e2_{s+1}}$.
   Go to stage $s+1$.

End $\{$stage $s\}$

We now prove that each of the invariants, (H1)–(H6), are satisfied by the construction. To begin with, it is easy to verify that (H2), (H3), (H6) are satisfied. (H2) follows from the enumeration in step 7 of the construction. (H3) is an immediate consequence of property (c) of Similar. (H6) follows from the definitions of $m1_s, m2_s$ and n_s.

We show that (H1), (H4), and (H5) hold by induction. We assume that (H1)–(H6) hold for $s=t$. We now show that they also hold for $s=t+1$. In the sequel, we use (Hi) $(s=u)$ to denote invariant (Hi), with s replaced by u. We consider two cases.

Case 1: $P1_{t+1}\cap P1_t\neq\emptyset$ and $P1_{t+1}\cap P2_t\neq\emptyset$.

We first show that $\bigcup_{i\in P1_{t+1}}[W_{p_i,m1_{t+1}}]\supseteq L^t_{e1_t}\cup L^t_{e2_t}$. From (H1) $(s=t)$, we get $\bigcup_{i\in P1_t}[W_{p_i,m1_t}]\subseteq\bigcap_{i\in P1_t}[W_{p_i,n_t}]$. Hence, for each $k\in P1_t$, $L^t_{e1_t}\subseteq W_{p_k,n_t}$. Let $k'\in(P1_{t+1}\cap P1_t)$ (such a k' exists since $P1_{t+1}\cap P1_t\neq\emptyset$). Clearly, $L^t_{e1_t}\subseteq W_{p_{k'},n_t}$. But (H6) $(s=t)$ implies that $m1_{t+1}>n_t$; hence $L^t_{e1_t}\subseteq W_{p_{k'},m1_{t+1}}\subseteq\bigcup_{i\in P1_{t+1}}[W_{p_i,m1_{t+1}}]$. Now, we show that $L^t_{e2_t}\subseteq\bigcup_{i\in P1_{t+1}}[W_{p_i,m1_{t+1}}]$. By (H4) $(s=t)$, it is sufficient to prove that $\bigcup_{i\in P2_t}[W_{p_i,m2_t}]\subseteq\bigcup_{i\in P1_{t+1}}[W_{p_i,m1_{t+1}}]$. (H3) $(s=t)$ implies that $\bigcup_{i\in P2_t}[W_{p_i,m2_t}]\subseteq\bigcap_{i\in P2_t}[W_{p_i,n_t}]$. Hence, for each $k\in P2_t$, $\bigcup_{i\in P2_t}[W_{p_i,m2_t}]\subseteq W_{p_k,n_t}$. Let $k'\in P1_{t+1}\cap P2_t$ (such a k' exists since $P1_{t+1}\cap P2_t\neq\emptyset$). Clearly, $\bigcup_{i\in P2_t}[W_{p_i,m2_t}]\subseteq W_{p_{k'},n_t}\subseteq W_{p_{k'},m1_{t+1}}$ (since (H6) $(s=t)$ implies that $m1_{t+1}>n_t$). Therefore, $L^t_{e2_t}\subseteq\bigcup_{i\in P1_{t+1}}[W_{p_i,m1_{t+1}}]$.

We now prove (H1) $(s=t+1)$. Step 7 in the construction ensures $\bigcup_{i\in P1_{t+1}}[W_{p_i,m1_{t+1}}]\subseteq L^{t+1}_{e1_{t+1}}$ (note that this is the only place where something is enumerated in $L^{t+1}_{e1_{t+1}}$ in stage t). Now, since $e1_{t+1}$ is either $e1_t$ or $e2_t$, and $\bigcup_{i\in P1_{t+1}}[W_{p_i,m1_{t+1}}]\supseteq L^t_{e1_t}\cup L^t_{e2_t}$, we have $\bigcup_{i\in P1_{t+1}}[W_{p_i,m1_{t+1}}]\supseteq L^{t+1}_{e1_{t+1}}$. Thus, (H1) $(s=t+1)$ holds.

To see that (H4) $(s=t+1)$ holds, it is sufficient to observe that $L^{t+1}_{e2_{t+1}}-\bigcup_{i\in P2_{t+1}}[W_{p_i,m2_{t+1}}]\subseteq L^t_{e1_t}\cup L^t_{e2_t}\subseteq L^{t+1}_{e1_{t+1}}$ (by argument in the proof of (H1) $(s=t+1)$).

To show (H5) ($s=t+1$), we first observe that the intersection of $P1_{t+1}$ and Q is at most $\frac{1}{2}w$, where $Q=P2_t$ if $e2_t=e2_{t+1}$, $Q=P1_t$ otherwise. This observation together with $\bigcup_{i\in P1_{t+1}}[W_{p_i,m1_{t+1}}]\supseteq L^t_{e1_t}\cup L^t_{e2_t}$ and (H1)–(H5) ($s=t$) imply that the number of grammars in $p_1,p_2,\ldots,p_r$ which enumerate any element in $L^t_{e2_{t+1}}$ is at least $\frac{3}{2}w$. Thus, (H5) ($s=t+1$) immediately follows.

Case 2: $P1_{t+1}\cap P1_t=\emptyset$ or $P1_{t+1}\cap P2_t=\emptyset$.

In this case we show that (H1) ($s=t+1$) holds. There are two subcases.

Subcase a: $e1_{t+1}=e1_t$.

Since $e1_{t+1}=e1_t$, it is sufficient to show that $L^t_{e1_t}\subseteq\bigcup_{i\in P1_{t+1}}[W_{p_i,m1_{t+1}}]$ (since step 7 in the construction guarantees that $\bigcup_{i\in P1_{t+1}}[W_{p_i,m1_{t+1}}]\subseteq L^{t+1}_{e1_{t+1}}$). Now, (H1) ($s=t$) implies that $L^t_{e1_t}\subseteq\bigcap_{i\in P1_t}[W_{p_i,n_t}]$. Hence, for each $k\in P1_t$, $L^t_{e1_t}\subseteq W_{p_k,n_t}$. Also, since $e1_{t+1}=e1_t$, we have $P1_{t+1}\cap P1_t\neq\emptyset$. Let $k'\in P1_{t+1}\cap P1_t$. Clearly, $L^t_{e1_t}\subseteq W_{p_{k'},n_t}\subseteq W_{p_{k'},m1_{t+1}}\subseteq\bigcup_{i\in P1_{t+1}}[W_{p_i,m1_{t+1}}]$.

Subcase b: $e1_{t+1}=e2_t$.

Again, step 7 in the construction guarantees that $\bigcup_{i\in P1_{t+1}}[W_{p_i,m1_{t+1}}]\subseteq L^{t+1}_{e1_{t+1}}$. Now suppose by way of contradiction, $(\exists x)[x\in L^{t+1}_{e1_{t+1}}-\bigcup_{i\in P1_{t+1}}[W_{p_i,m1_{t+1}}]]$. Clearly, $x\in L^t_{e2_t}$. But, (H5) ($s=t$) implies that $(\forall x\in L^t_{e2_t})[\mathrm{card}(\{j\in\{1,2,\ldots,r\}-P1_t\mid x\in W_{p_j,n_t}\})\geqslant\frac{1}{2}w]$. But since, $P1_{t+1}\cap P1_t=\emptyset$, there exists at least one $i\in P1_{t+1}$ such that $x\in W_{p_i,m1_{t+1}}$ – a contradiction. Hence, $L^{t+1}_{e1_{t+1}}=\bigcup_{i\in P1_{t+1}}[W_{p_i,m1_{t+1}}]$.

We leave details of the proof of (H4) and (H5). It should be noted that they immediately hold if the first **if** in step 6 in the construction succeeds; otherwise they can be shown to hold using (H1) ($s=t$), (H3) ($s=t$), (H4) ($s=t$), and (H5) ($s=t$).

We now show how the invariants imply the lemma.

Suppose there is exactly one language, L, which has at least w grammars in the set $\{p_1,\ldots,p_r\}$. In this case clearly, $m1_s$ is unbounded and by (H1), at least one of L_1 and L_2 is the same as L (depending on whether $e1_s$ takes value 1 or 2 infinitely often).

Suppose there are two distinct languages L and L' which have at least w grammars in the set $\{p_1,\ldots,p_r\}$. It is easy to see that both $m1_s$ and $m2_s$ are unbounded and, for all but finitely many s, $[P1_{s+1}\cap P1_s=\emptyset\vee P1_{s+1}\cap P2_s=\emptyset]$. It now follows using (H1), (H3), and (H5) that both L_1 and L_2 belong to $\{L,L'\}$ and are distinct.

Thus, $(\forall L)[\mathrm{card}(\{i\mid 1\leqslant i\leqslant r\wedge W_{p_i}=L\})\geqslant w\Rightarrow L=L_1\vee L=L_2]$. □

Theorem 10. $(\forall m,n\mid m/n>\frac{5}{8})[\mathbf{Team}^m_n\mathbf{TxtFex}_2=\mathbf{TxtFex}_2]$.

Proof. This proof uses Lemma 4 which shows that there exist recursive functions G_1 and G_2, such that for any set S of r grammars, $(\forall L\mid\mathrm{card}(\{i\in S\mid W_i=L\})>\frac{2}{5}r)$ $[W_{G_1(S)}=L]\ \vee\ [W_{G_2(S)}=L]$.

Let m,n be as described in the hypothesis of the theorem. Suppose a team of n machines, $\mathbf{M}_1,\mathbf{M}_2,\ldots,\mathbf{M}_n$, are given. We describe a machine $\mathbf{M}$ that $\mathbf{TxtFex}_2$-identifies any language which is $\mathbf{Team}^m_n\mathbf{TxtFex}_2$-identified by the team consisting of machines $\mathbf{M}_1,\mathbf{M}_2,\ldots,\mathbf{M}_n$.

Suppose the team consisting of machines $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n$ $\mathbf{Team}_n^m\mathbf{TxtFex}_2$-identifies L. Let T be any text for L. Without loss of generality, we assume that for $1 \leqslant j_1 < j_2 \leqslant n$, $\mathrm{LastGram}_2(\mathbf{M}_{j_1}, T)$ and $\mathrm{LastGram}_2(\mathbf{M}_{j_2}, T)$ (if defined) are disjoint (this can easily be ensured by padding). This assumption is only for the ease of presentation of the proof. For $l \in N$, let S_l denote the lexicographically least subset of $\{1, \ldots, n\}$ of cardinality m such that $\max(\{\mathrm{LastMindChange}_2(\mathbf{M}_j, T[l]) \mid j \in S_l\})$ is minimized. Note that $\lim_{l\to\infty} S_l$ exists (since the team consisting of machines $\mathbf{M}_1, \ldots, \mathbf{M}_n$ $\mathbf{Team}_n^m\mathbf{TxtFex}_2$-identifies L). Let $S = \lim_{l\to\infty} S_l$.

For $l \in \mathbb{N}$, let $X_l = \bigcup_{j\in S_l}[\mathrm{LastGram}_2(\mathbf{M}_j, T[l])]$. Since, for each $j \in S$, $\mathbf{M}_j$ converges on T to a set of at most 2 grammars, $\lim_{l\to\infty} X_l$ exists – let this limit be X. Moreover, $\mathrm{card}(X) \leqslant 2m$ and at least $m-(n-m)$ of the grammars in X are grammars for L (since the team consisting of machines $\mathbf{M}_1, \ldots, \mathbf{M}_n$ $\mathbf{Team}_n^m\mathbf{TxtFex}_2$-identifies L). Thus, at least $(2m-n)/2m$ (which is greater than $\frac{2}{5}$) fraction of grammars in X are for L. This, together with Lemma 4, implies that at least one of $G_1(X)$ and $G_2(X)$ is a grammar for L.

We now describe the behavior of our machine $\mathbf{M}$. For $n \in \mathbb{N}$, $\mathbf{M}(T[n]) = \mathrm{BestGram}(\{G_1(X_n), G_2(X_n)\}, T[n])$. It is easy to see from the analysis on X above and the property of function BestGram (Lemma 2) that $\mathbf{M}$ $\mathbf{TxtFex}_2$-identifies L. □

Theorem 11. $\mathbf{Team}_6^5\mathbf{TxtFex}_2 - \mathbf{TxtFex}_2 \neq \emptyset$.

Proof. Consider the following class of languages.

$\mathcal{L} = \{L \mid \mathrm{card}(\{w \leqslant 5 \mid (\exists x \leqslant 1)[\mathrm{card}(\{\langle 2w, y\rangle \mid y \in \mathbb{N}\} \cap L) < \infty \wedge \mathrm{card}(\{\langle 2w+1, y\rangle \mid y \in \mathbb{N}\} \cap L) < \infty \wedge W_{\max(\{y \mid \{\langle 2w+x, y\rangle \mid y\in\mathbb{N}\}\cap L\})} = L]\}) \geqslant 5\}$.

We now show that $\mathcal{L} \in \mathbf{Team}_6^5\mathbf{TxtFex}_2$. Consider a team of six machines $\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_5$ such that machine $\mathbf{M}_i$, $0 \leqslant i \leqslant 5$, behaves as follows on any text T.

Begin $\{\mathbf{M}_i(T[n])\}$
  if $\{y \mid \langle 2i, y\rangle \in \mathrm{content}(T[n])\} \neq \emptyset$
  then
    let $m_1 = \max(\{y \mid \langle 2i, y\rangle \in \mathrm{content}(T[n])\})$
  else let $m_1 = 0$.
  endif
  if $\{y \mid \langle 2i+1, y\rangle \in \mathrm{content}(T[n])\} \neq \emptyset$
  then
    let $m_2 = \max(\{y \mid \langle 2i+1, y\rangle \in \mathrm{content}(T[n])\})$
  else let $m_2 = 0$.
  endif
  Output $\mathrm{BestGram}(m_1, m_2, T[n])$.
End $\{\mathbf{M}_i(T[n])\}$

It is easy to verify that the team consisting of machines, $\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_5$, $\mathbf{Team}_6^5\mathbf{TxtFex}_2$-identifies $\mathcal{L}$.

We now show that $\mathscr{L} \notin \mathbf{TxtFex}_2$. Suppose by way of contradiction that $\mathbf{M}$ $\mathbf{TxtFex}_2$-identifies $\mathscr{L}$. We then show that there exists a language in $\mathscr{L}$ that $\mathbf{M}$ fails to $\mathbf{TxtFex}_2$-identify. The description of this witness proceeds in stages and uses the operator recursion theorem [4]. The construction is somewhat on the lines of the diagonalization argument presented in our proof of Theorem 8(a). We give an informal description of the idea first.

At each stage s, the construction makes use of initial sequence σ_s. By the use of the operator recursion theorem, we initialize σ_0 to "agree" with languages in $\mathscr{L}$. We then proceed in stages. At each stage s, an attempt is made to find a sequence τ extending σ_s such that $\mathbf{M}$ undergoes a mind change on τ with respect to $\mathbf{TxtFex}_2$-identification. If such an attempt is successful at every stage then the construction yields a language in $\mathscr{L}$ for which $\bigcup_{s \in N} \sigma_s$ is a text and on this text $\mathbf{M}$ does not converge to up to two grammars. If on the other hand, an attempt to find a mind change is unsuccessful at some stage s then the machine $\mathbf{M}$ has essentially locked itself to a set of up to two grammars on all suitable extensions of σ_s. The construction then describes a number of languages in $\mathscr{L}$ which diagonalize against the grammars on which $\mathbf{M}$ has become locked. We now proceed formally.

By the operator recursion theorem, there exists a 1–1, recursive, increasing function p, such that the languages $W_{p(i)}$ can be described as follows.

Enumerate $\langle i, p(i) \rangle$ in $W_{p(j)}$ for $i \leqslant 9$ and $j \leqslant 9$. Let $W^s_{p(\cdot)}$ denote $W_{p(\cdot)}$ enumerated before state s. Let $\mathbf{Last}_2(\sigma) = \mathrm{LastGram}_2(\mathbf{M}, \sigma)$. (For ease of construction we assume without loss of generality that $\mathbf{Last}_2(\sigma)$ is always of cardinality 2.) Let σ_0 be such that $\mathrm{content}(\sigma_0) = \{\langle i, p(i) \rangle \mid i \leqslant 9\}$. Go to stage 0.

Stage s

Dovetail steps 1 and 2 until, if ever, step 1 succeeds. If and when step 1 succeeds, go to step 3.

1. Search for an extension τ of σ_s such that $\mathrm{content}(\tau) - \mathrm{content}(\sigma_s) \subseteq \{\langle x, y \rangle \mid x > 9\}$, such that $\mathbf{Last}_2(\tau) \neq \mathbf{Last}_2(\sigma_s)$.
2. Let $m_1 = 1 + \max(\{x \mid (\exists y)[\langle x, y \rangle \in W_{p(0)}$ enumerated till now$]\})$.
   Let $r_1 = 3 + \max(\{y \mid (\exists x \leqslant 11)[\langle x, p(y) \rangle \in W_{p(0)}$ enumerated till now$]\})$.
   2.1. Enumerate $\langle 10, p(r_1) \rangle$ in $W_{p(0)}$.
   Enumerate $W_{p(0)}$ enumerated till now in $W_{p(i)}$, $i \leqslant 9$ and $W_{p(r_1)}$, $W_{p(r_1+1)}$, $W_{p(r_1+2)}$.
   Enumerate $\langle m_1, 0 \rangle$ in $W_{p(0)}$, $W_{p(2)}$, $W_{p(4)}$, $W_{p(6)}$ and $W_{p(r_1)}$.
   Search for a $q \in \mathbf{Last}_2(\sigma_s)$, such that W_q enumerates $\langle m_1, 0 \rangle$.
   If and when the search succeeds, go to step 2.2.
   2.2. Enumerate $\langle 10, p(r_1+1) \rangle$ in $W_{p(i)}$, $i \leqslant 9$ and $W_{p(r_1+1)}$, $W_{p(r_1+2)}$.
   Enumerate $\langle m_1+1, 0 \rangle$ in $W_{p(3)}$, $W_{p(5)}$, $W_{p(7)}$, $W_{p(9)}$, $W_{p(r_1+1)}$.
   Search for $q' \in \mathbf{Last}_2(\sigma_s) - \{q\}$, such that $W_{q'}$ enumerates $\langle m_1+1, 0 \rangle$.
   If and when the search succeeds, go to step 2.3.

2.3. Enumerate $\langle m_1, 0\rangle$ and $\langle m_1+1, 0\rangle$ in $W_{p(0)}$, $W_{p(2)}$, $W_{p(7)}$, $W_{p(9)}$, $W_{p(r_1+1)}$.
Search for a $q'' \in \mathbf{Last}_2(\sigma_s)$, such that both $\langle m_1, 0\rangle$ and $\langle m_1+1, 0\rangle$ are enumerated in $W_{q'}$.
If and when the search succeeds go to step 2.4.

2.4. Let $x \in \{m_1, m_1+1\}$ be such that all grammars in $\mathbf{Last}_2(\sigma_s)$ enumerate $\langle x, 0\rangle$.
Let x' be the only element in $\{m_1, m_1+1\} - \{x\}$.
Enumerate $\langle 10, p(r_1+2)\rangle$ in $W_{p(i)}$, $i \leqslant 9$ and $W_{p(r_1+2)}$.
Enumerate $\langle x', 0\rangle$ in $W_{p(1)}$, $W_{p(8)}$ and $W_{p(r_2+1)}$.
Note that, if the search in step 1 does not succeed, then either $W_{p(4)}$ and $W_{p(6)}$ or $W_{p(3)}$ and $W_{p(5)}$ are the same as $W_{p(1)}$.

3. Let $S = \text{content}(\tau) \cup \bigcup_{i \leqslant 9} [W_{p(i)}$ enumerated till now].
Enumerate S in $W_{p(i)}$, $i \leqslant 9$.
Let σ_{s+1} be an extension of τ such that $\text{content}(\sigma_{s+1}) = S$.
Go to stage $s+1$.
End stage s

Now consider the following cases.

Case 1: All stages halt.
In this case let $L = W_{p(0)}$. It is easy to see that $L \in \mathscr{L}$. However, $\mathbf{M}$ on $\bigcup_s \sigma_s$, a text for L, does not converge to at most two grammars.

Case 2: Stage s starts but does not halt.

If the search in step 2.1 does not succeed, then let $L = W_{p(0)}$. If the search in step 2.1 succeeds, but the search in step 2.2 fails, then let $L = W_{p(3)}$. If the search in step 2.1 and 2.2 succeed, but the search in step 2.3 fails, then let $L = W_{p(0)}$. If the search in step 2.1, 2.2 and 2.3 succeed, then let $L = W_{p(1)}$. It is easy to see that in all these three cases, $L \in \mathscr{L}$ and $L \notin \{W_q \mid q \in \mathbf{Last}_2(\sigma_s)\}$. Thus we have that $\mathscr{L} \not\subseteq \mathbf{TxtFex}_2(\mathbf{M})$.

Thus we have that $\mathscr{L} \notin \mathbf{TxtFex}_2$. $\square$

3.7. *Aggregation for language identification from informants*

Results presented in the previous section were for language learning criteria in which learning takes place from positive data only. In the present section, we record similar results for learning criteria in which learning takes place from both positive and negative data. It should be noted that the proof techniques for language learning from informants and function learning from graphs are very similar. This is despite the fact that identification of recursively enumerable languages from informants differs from identification of recursive functions because a learning machine is required to converge to a total program in identifying recursive functions whereas a machine identifying recursively enumerable languages from informants converges to grammars (which are semi-decision procedures).

Identification from texts is an abstraction of learning from positive data. Similarly, learning from both positive and negative data can be abstracted as identification from informants. The notion of informants, defined below, was first considered by Gold [13].

Definition 18. A text I is called an *informant* for a language L just in case content$(I)=\{\langle x,1\rangle \mid x\in L\}\cup\{\langle x,0\rangle \mid x\notin L\}$.

The next definition formalizes identification in the limit from informants.

Definition 19. (a) $\mathbf{M}$ $\mathbf{InfEx}$-*identifies* L (written: $L\in\mathbf{InfEx}(\mathbf{M})$) $\Leftrightarrow$ ($\forall$ informants I for L)$(\exists i \mid W_i=L)(\overset{\infty}{\forall} n)[\mathbf{M}(I[n])=i]$.

(b) $\mathbf{InfEx}=\{\mathscr{L}\subseteq\mathscr{E} \mid (\exists\mathbf{M})[\mathscr{L}\subseteq\mathbf{InfEx}(\mathbf{M})]\}$.

We leave it to the reader to similarly define **InfFin**, **InfBc**, and for each $b\in\mathbb{N}^+\cup\{*\}$, $\mathbf{InfFex}_b$. Also, for $m,n\in\mathbb{N}^+$ and for each $\mathbf{I}\in\{\mathbf{InfFin}, \mathbf{InfEx}, \mathbf{InfFex}_b, \mathbf{InfBc}\}$, we can define $\mathbf{Team}^m_n\mathbf{I}$-identification. We now present aggregation results for these new criteria.

For finite identification from informants, the aggregation ratio is $\frac{2}{3}$ as implied by the following results. This is not unexpected given results about finite function identification and finite language identification from texts.

Theorem 12. (a) $(\forall m,n\in\mathbb{N}^+ \mid m/n>\frac{2}{3})$ $[\mathbf{Team}^m_n\mathbf{InfFin}=\mathbf{InfFin}]$.

(b) $\mathbf{InfFin}\subset\mathbf{Team}^2_3\mathbf{InfFin}$.

Proof. Part (b) can be obtained as a corollary to the corresponding function learning result. For part (a), suppose $\mathbf{M}_1,\mathbf{M}_2,\ldots,\mathbf{M}_n$ and an informant T are given. Let s_T be the least number if any such that there exists a set $S\subseteq\{1,\ldots,n\}$ of cardinality m, such that, for each $j\in S$, $\mathbf{M}_j(T[s_T])\neq\perp$. Then $\mathbf{M}(T[s])=\perp$ for $s<s_T$, and, for $s\geqslant s_T$, $\mathbf{M}(T[s])=i$, where i is such that $W_i=\{x \mid \mathrm{card}(\{j\in S \mid x\in W_{\mathbf{M}_j(T[s_T])}\})\geqslant 2m-n\}$. It is easy to verify that $\mathbf{M}$ **InfFin**-identifies any language that is $\mathbf{Team}^m_n\mathbf{InfFin}$-identified by $\mathbf{M}_1,\mathbf{M}_2,\ldots,\mathbf{M}_n$. $\square$

For identification in the limit, however, aggregation turns out to be different for informants and texts. In fact language identification from informants behaves very much like function learning, as aggregation ratio for **InfEx** is $\frac{1}{2}$. Also, the aggregation ratio for **InfBc** is $\frac{1}{2}$. These observations are summarized in the following result.

Theorem 13. *Let* $\mathbf{I}\in\{\mathbf{InfEx}, \mathbf{InfBc}\}$.

(a) $(\forall m,n \mid m/n>\frac{1}{2})[\mathbf{Team}^m_n\mathbf{I}=\mathbf{I}]$.

(b) $\mathbf{I}\subset\mathbf{Team}^1_2\mathbf{I}$.

Proof. Part (b) can be proved using the language learning analog of the proof used to show $\mathbf{I}\subset\mathbf{Team}^1_2\mathbf{I}$ for $\mathbf{I}\in\{\mathbf{Ex},\mathbf{Bc}\}$. For part (a) suppose $m>\frac{1}{2}n$. $\mathbf{Team}^m_n\mathbf{InfEx}\subseteq\mathbf{InfEx}$ can be obtained as a corollary to Theorem 14 below (since, for $m>\frac{1}{2}n$, $\mathbf{Team}^m_n\mathbf{InfEx}\subseteq\mathbf{InfFex}_m$; proof similar to proofs for Theorems 6(a) and 8(b)). Essentially the proof of $\mathbf{Team}^m_n\mathbf{TxtBc}\subseteq\mathbf{TxtBc}$ can also be used to show that $\mathbf{Team}^m_n\mathbf{InfBc}\subseteq\mathbf{InfBc}$. $\square$

Table 1

Type of identification	Finite	Limit	Vacillatory				Behaviorally correct
			2	3	...	*	
Function (graph)	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
Language (text)	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{5}{6}$	?	?	$\frac{1}{2}$	$\frac{1}{2}$
Language (informant)	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$

Theorem 14 can be proved using techniques similar to that used by Case and Smith [8] to show that $\mathbf{Fex} = \mathbf{Ex}$.

Theorem 14. $(\forall b \in \mathbb{N}^+ \cup \{*\})[\mathbf{InfFex}_b = \mathbf{InfEx}]$.

Hence, Theorem 13 holds for vacillatory identification from informants, too.

4. Conclusion

Clearly, aggregation issues for $\mathbf{TxtFex}_b$, where $b \neq * \wedge b > 2$, are open. Only partial results can be shown at this stage, as the combinatorial complexity of the simulation arguments become difficult to handle. We summarize the state of art about aggregation ratios in Table 1; the symbol ? denotes open questions.

Acknowledgement

A preliminary version of this paper was presented at the *Fourth International Workshop on Algorithmic Learning Theory* (ALT-93, Tokyo, November 1993). We wish to thank Frank Stephan, the reviewers of ALT-93, and the reviewers of this journal for several helpful comments.

References

[1] J.M. Barzdin, Two theorems on the limiting synthesis of functions, in: *Theory of Algorithms and Programs*, Vol. 210 (Latvian State University, Riga 1974) 82–88 (in Russian).

[2] J.M. Barzdin and K. Podnieks, The theory of inductive inference, in: *Mathematical Foundations of Computer Science*, (High Tatras, Czechoslovakia, 1973) 9–15.

[3] M. Blum, A machine independent theory of the complexity of recursive functions, *J. ACM* **14** (1967) 322–336.

[4] J. Case, Periodicity in generations of automata, *Math. Systems Theory* **8** (1974) 15–32.

[5] J. Case, The power of vacillation, in: D. Haussler and L. Pitt, ed., *Proc. Workshop on Computational Learning Theory* (Morgan Kaufmann, Los Altos, CA, 1988); 133–142; expanded in [6].

[6] J. Case, The power of vacillation in language learning, Tech. Report 93–08, University of Delaware, 1992; expands on [5]; journal article under review.
[7] J. Case and C. Lynes, Machine inductive inference and language identification, in: M. Nielsen and E.M. Schmidt, ed., *Proc. 9th Internat. Colloq. on Automata, Languages and Programming*, Lecture Notes in Computer Science Vol. 140 (Springer, Berlin, 1982) 107–115.
[8] J. Case and C. Smith, Comparison of identification criteria for machine inductive inference, *Theoret. Comput. Sci.* **25** (1983) 193–220.
[9] R.P. Daley, Inductive inference hierarchies: probabilistic vs pluralistic, in: *Mathematical Methods of Specification and Synthesis of Software Systems, Wendisch-Rietz GDR*, Lecture Notes in Computer Science, Vol. 215 (Springer, Berlin, 1985) 73–82.
[10] R.P. Daley, B. Kalyanasundaram and M. Velauthapillai, Breaking the probability $\frac{1}{2}$ barrier in fin-type learning, in: *Proc. 5th Ann. Workshop on Compuptational Learning Theory, Pittsburgh, Pennsylvania* (ACM, New York, 1992) 203–217.
[11] R.P. Daley, L. Pitt, M. Velauthapillai and T. Will, Relations between probabilistic and team one-shot learners, in: L. Valiant and M. Warmuth, ed., *Proc. Workshop on Computational Learning Theory* (Morgan Kaufmann, Los Altos, CA, 1991) 228–239.
[12] R. Freivalds, Functions computable in the limit by probabilistic machines, in: *Mathematical Foundations of Computer Science* (1974) 77–87.
[13] E.M. Gold, Language identification in the limit, *Inform. and Control* **10** (1967) 447–474.
[14] J. Hopcroft and J. Ullman, *Introduction to Automata Theory Languages and Computation* (Addison-Wesley, Reading, MA, 1979).
[15] S. Jain and A. Sharma, Finite learning by a team, in: M. Fulk and J. Case, eds., *Proc. 3 Ann. Workshop on Computational Learning Theory, Rochester, New York* (Morgan Kaufmann, Los Altos, CA, 1990).
[16] S. Jain and A. Sharma, Language learning by a team, in: M.S. Paterson, ed, *Proc. 17th Internat. Colloq. on Automata, Languages and Programming*, Lecture Notes in Computer science, Vol. 443 (Springer, Berlin, 1990) 153–166.
[17] S. Jain and A. Sharma, Computational limits on team identification of languages, Tech. Report 9301, School of Computer Science and Engineering, University of New South Wales, 1993.
[18] S. Jain and A. Sharma, Probability is more powerful than team for language identification, in: *Proc. 6th Ann. Conf. Computational Learning Theory, Santa Cruz, California* (ACM, New York, 1993) 192–198.
[19] M. Machtey and P. Young, *An Introduction to the General Theory of Algorithms* (North-Holland, New York, 1978).
[20] D. Osherson, M. Stob and S. Weinstein, Aggregating inductive epertise, *Inform. and Control.* **70** (1986) 69–95.
[21] D. Osherson and S. Weinstein, Criteria of language learning, *Inform. and Control*, **52** (1982) 123–138.
[22] L. Pitt, A characterization of probabilistic inference, Ph. D. Thesis, Yale University, 1984.
[23] L. Pitt and C. Smith, Probability and plurality for aggregations of learning machines, *Inform. and Comput.* **77** (1988) 77–92.
[24] H. Rogers, Gödel numberings of partial recursive functions, *J. Symbolic Logic* **23** (1958) 331–341.
[25] H. Rogers, *Theory of Recursive Functions and Effective Computability*, McGraw-Hill, New York, 1967); repreinted, MIT Press Cambridge, MA, 1987.
[26] C. Smith, The power of pluralism for automatic program synthesis, *J. ACM* **29** (1982) 1144–1165.
[27] M. Velauthapillai, Inductive inference with bounded number of mind changes, in: *Proc. 2nd Annual Workshop on Computational Learning Theory, Santa Cruz, California* (Morgan Kaufmann, Los Altos, CA (1989) 200–213.